UPPSALA
UNIVERSITET

IT 23 042

Degree project 30 credits

June 2023

Exploration-Exploitation Trade-off Approaches in Multi-Armed Bandit

Duc Huy Le

Master's Programme in Data Science

UPPSALA
UNIVERSITET

Exploration-Exploitation Trade-off Approaches in Multi-Armed Bandit

Duc Huy Le

Abstract

Multi-armed bandit, a popular framework for sequential decision-making problems, has recently gained significant attention due to numerous applications. In Multi-armed Bandit, an agent faces the central challenge of choosing exploitation of its belief to hopefully gain a high reward and exploration to improve its knowledge of the environment, and any good strategy has to efficiently balance between the two actions. Being particularly interested in the Bernoulli reward signal, in this project, we studied two canonical algorithms, ε-greedy and Thompson Sampling, and proposed the Surprise-based Exploration policy, a novel method inspired by the intrinsic motivation in psychology, that can be incorporated in many existing strategies to promote exploration into less known actions. Our experimental results show that Thompson Sampling and our proposed policy are greatly efficient with a small set of arms, however, gradually drop when the number of arms increases. Besides, we tackled the challenge exploration with a large number of arms and introduced a solution that distributes a portion of the time horizon to reduce the arm set with our proposed Successive-Reject Best Arms Identification algorithm. The empirical studies show that our method significantly improves the performance of Thompson Sampling, however, it fails to give ε-greedy enhancements.

Faculty of Science and Technology

Uppsala University, Uppsala

Supervisor: Jens Sjölund Subject reader: André Teixeira

Examiner: Matteo Magnani

Acknowledgements

I would first like to express my sincere gratitude to my thesis advisors, Jens Sjölund and André Teixeira, for their invaluable support and insights throughout the project. Their expertise and dedication guided me through tough times toward the success of this thesis. The wonderful experiences working with them on this project will undoubtedly have a great impact on my future research career.

I would also like to thank Matteo Magnani and Yasser Kaddoura for providing detailed and constructive feedback, which helps me finish writing this thesis.

Finally, I am deeply grateful to my family and friends for their love and support during the process.

Contents

List of Abbreviations

RL	Reinforcement Learning
ML	Machine Learning
MAB	Multi-armed Bandit
IID	Independent and Identically Distributed
KL-divergence	Kullback-Liebler Divergence
SR	Successive-Reject

Chapter 1

Introduction

1.1 Motivation

In the realm of decision-making, one of the most challenging scenarios is when we encounter an unknown environment. This refers to circumstances where the decision-makers have to take actions without knowing the corresponding outcome and without complete information about the underlying environment. In such cases, the decision-makers are required to design strategies that could adapt over time and use the feedback from interacting with the environment to explore hidden patterns and refine their decision-making process and maximise overall gains.

Multi-armed Bandit (MAB), the simplest form of Reinforcement Learning (RL), is a popular framework for sequential decision-making problems with an unknown environment, which despite already being introduced in 1952 by Herbert Robbins[14], has just received notable attention. The MAB framework model a flexible and powerful approach in numerous applications in different fields, such as online advertising, recommendation systems, resource allocation and clinical trials. For example, a study in 2010[22] discussed the application of MAB models in clinical trial design and showed significant benefits in resource allocation and effective treatment identification. In the same year, another study[11] proposed a contextual MAB algorithm for personalised news article recommendation that takes into account the user's historical behaviour and contextual information, such as location and time of the day, to suggest news articles that are likely to be of interest to the user. In a 2021 publication[10], the authors presented a novel approach for solving the egress node selection problem in large-scale networks using MAB techniques.

1.2 Problem formulation

A key concept in MAB is the *trade-off between exploration and exploitation.* With the goal to maximise reward getting over time, on the one hand, the MAB wants to exclusively perform the action with the highest observed reward, but it may miss the potentially better actions due to its incorrect knowledge of the environment. On the other hand, if the agent focuses on exploring less visited arms to gather more information about the underlying environment, it can waste time and resources and finally fail in the goal of achieving a high reward. The exploration-exploitation trade-off balancing is the central challenge and has been extensively studied in the literature of MAB with many algorithms proposed to tackle the problem.

This project focuses on addressing the following three research questions on the premise of the stochastic MAB model where the reward for each action is sampled from a Bernoulli distribution.

RQ1) What are the potential solution candidates to handle the exploration-exploitation trade-off in MAB problem with Bernoulli reward signal?
RQ2) What is the empirical performance of the solution candidates in different scenarios of the MAB environment?
RQ3) How does the arm reduction method improve the performance of the standard MAB algorithms in the extreme case of numerous arms?

Our contribution is threefold, characterized by carrying out the above questions. Firstly, we investigate two canonical solutions, ε-greedy and Thompson Sampling and propose a new exploration policy called Surprise-based exploration as the answer to RQ1. For RQ2, we design and conduct experiments with several MAB environment scenarios to examine the performance of the three proposed candidates. Lastly, we address the issue of MAB problems with a large number of arms by proposing a method aimed at eliminating poor arms and analyse its performance through experiments.

1.3 Outline

The structure of this report is as follows. Chapter 2 provides an overview of the relevant background literature, covering the topic of RL and MAB. In chapter 3, we formulate our MAB problem and pose our first research question, which is answered in the remaining sections of the chapter by presenting two canonical methods and proposing the Surprise-based exploration policy. After that, we reintroduce our second research question that is investigated in chapter 4. Chapter 5 states the problem of many arms in MAB, proposes our solution and the third research question, which is addressed in chapter 6. Finally, chapter 7 provides a conclusion of the project, including achievements and limitations, and proposes potential future research directions.

Chapter 2

Background

In this chapter, we introduce the fundamental concepts relevant to this project, starting by giving an overview of reinforcement learning, a broader topic of Multi-armed Bandit, which is generally discussed afterwards. The content of this chapter is largely based on Chapters 1 and 4 of [18] and Chapter 1 of [17].

2.1 Reinforcement Learning

Reinforcement Learning (RL) is one of the major branches of machine learning (ML) that focuses on training agents to make decisions while interacting with complex environments, the underlying dynamic of which is fully or partially unknown to the agents. Here we introduce the notion and discuss the key elements of RL.

2.1.1 Overview

The idea that we learn by interacting with our surroundings is possibly the most intuitive notion considering the nature of learning. A toddler, for instance, learning how to walk, though without any explicit instructions on how to balance and move without falling, can develop a sense of how to shift their weight and move their legs in a coordinated manner through trial and error. They learn from observing the consequences of their actions such as falling, or a successful step and adjust their behaviour accordingly to improve their walking ability. Gradually, the child becomes more proficient with greater control over their movements, all through the process of interacting with the surrounding environment. Regardless of learning to walk, drive or even fly, we are keenly aware of how the environment reacts to our actions and strive to influence the outcomes through our behaviour.

Reinforcement Learning is foundationally inspired by the idea of learning from interaction, much more than any other approach to machine learning. Formally, RL can be defined as a computational framework for decision-making under uncertainty, where a goal-directed agent learns to take actions to maximise cumulative rewards over time, based on the feedback from the environment in the form of rewards or penalties, which could also be considered as negative rewards. The agent interacts with the environment in discrete time steps, at each step, it selects an action based on its current state and a policy that maps states to actions. At each time step, the agent performs an action and the environment transitions to a new state and returns a reward to the agent based on the state and its action. This closed-loop relationship between the agent and the environment is displayed in Figure 2.1. The agent's ultimate objective is to learn a policy that maximises the total reward it receives over time.

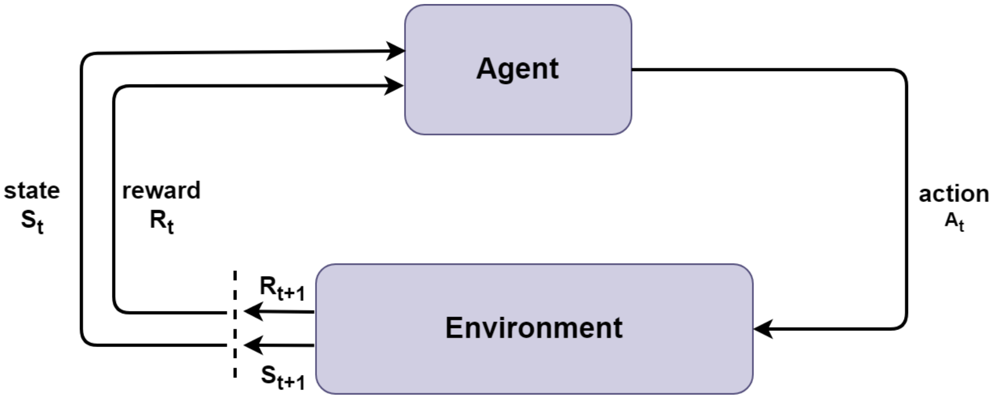


Figure 2.1: Diagram of RL. Reproduced from [18]

Compared to the other two more famous and studied genres of machine learning, supervised learning and unsupervised learning, RL is very different. Firstly, *supervised learning* learns from a training set of labelled samples provided by knowledgeable external supervisors with each sample comprising a situation description and a specification, or label, of the correct action the system should take in response to that situation. On the other hand, in RL problems, it is impractical to acquire examples of desired behavior that are both correct and representative of all the situations that the agent is required to act but the agent must be able to learn from its own experience. Secondly, although it is tempting to think RL is being included in *unsupervised learning*, which also does not have examples of correct behaviour, the optimal goals of these two kinds of learning are well distinguished. While unsupervised learning aims to find the hidden structure of the data, RL is trying to maximise a reward signal. Despite the fact that uncovering the environment dynamic via the agent's experience could be useful for RL, but by itself does not address the agent's problem of maximising the cumulative reward. Therefore, alongside supervised learning and unsupervised learning, RL is considered the third paradigm of ML.

2.1.2 Rewards

Up until now, we have provided a general explanation of the agent's optimal goal, which is 'to maximise cumulative rewards over time' and we will now express this idea in a more formal manner. In RL, a *reward signal* is used to guide the agent toward desired behaviour. A reward is a numerical value that the agent receives after performing an action at each time step. It indicates how good or bad the action was and is used by the agent to update its policy. Using the trajectory τ, which results from the dynamic interaction between the agent and the environment, we can attain a subset that comprises solely the rewards acquired by the agent:

$$R_1, R_2, R_3, R_4, R_5, \ldots$$

The agent's ultimate goal is to maximise the total reward it receives over the long run, which could be either a finite or infinite sum of rewards depending on whether τ is a finite trajectory, which could be caused by a time step limit T or by one or more terminal states in the state space, or not.

2.1.3 Policies

One key element in RL is *policy*, which defines the learning agent's way of selecting action at any given time and is usually represented by π, which is essentially a way of mapping the environment states to actions that should be taken in those states. Depending on the situation, the policy can be a basic function or lookup table, or it may require a more complex computation, such as a search algorithm. We can now refine the objective of the agent to find the *optimal policy* π^* that represents the optimal way of choosing an action for every state of the environment.

2.1.4 The Trade-Off Between Exploration and Exploitation

In RL, one central challenge that does not occur in any other kind of learning is the balance between *exploration* of the environment and *exploitation* of the agent's knowledge to obtain the optimal cumulative reward. Generally, with the goal of maximising the cumulative reward, an RL agent should prioritise actions that have previously yielded positive results. However, in order to discover new effective actions, the agent must also explore and try less discovered options. It is impossible to exclusively pursue exploration or exploitation without failing to complete the task at hand. In the first case, where the agent only explores the environment, it always selects actions that give more information about the environment. Gradually, the agent would gain more knowledge about the environment and which actions yield the most rewards. However, it would never be motivated to actually select these optimal actions. Conversely, solely exploiting 'optimal' actions will be restricted to the policy or approximate action-value function that the agent was initially provided with. The agent would never be exposed to alternative actions that might prove to be better choices than its current selection. Either of these two strategies possibly results in suboptimal reward, which is opposite to the objective of the agent. Therefore, an ideal agent must be able to merge both methods to accomplish its goal of maximising the return.

2.2 Multi-armed Bandit

Having introduced the key concepts associated with RL, now we move on to introduce the simplest form of RL, Multi-armed Bandit (MAB), which is first introduced by Robbins [14] and is the main problem in our project.

2.2.1 Overview

Consider the following learning scenario, where we are presented with a set of options or actions, and we have to make a choice repeatedly. Following each selection, we receive a numerical reward from a probability distribution that is influenced by the action performed. Our goal is to maximise the total reward we receive over a specific time period, for example, 1000 action selections or *time steps*.

This learning problem so far is the original form of the *Multi-armed Bandit*, named after a stylized gambling scenario in which a player faces several slot machines, a.k.a. one-armed bandits, that appear identical but yield different payoffs (Figure 2.2). The player (*agent*) has to pull one machine (an *action*) at every time step and the rewards are the payoffs for hitting the jackpot. The player's objective is to get the highest amount of money received from the machine over a specific number of plays.

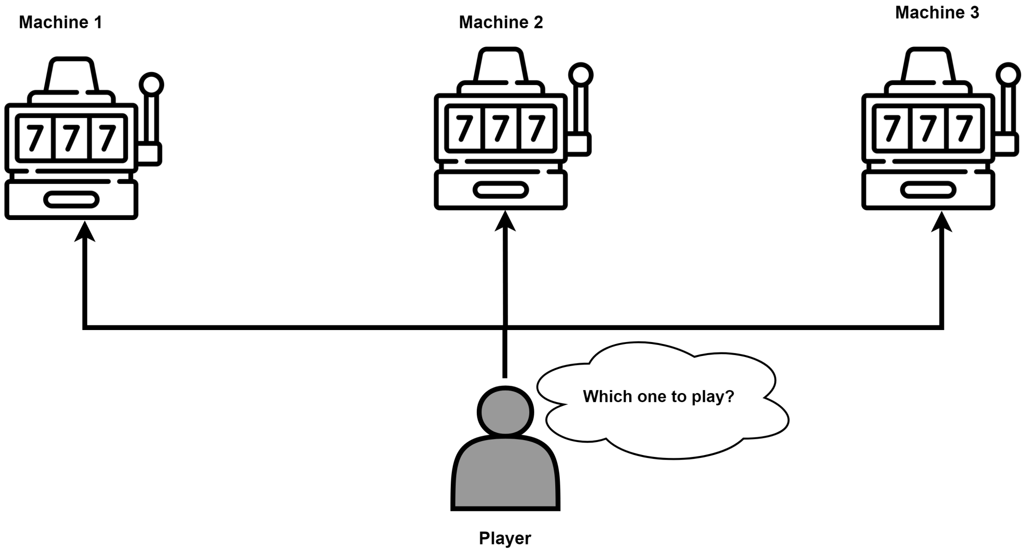


Figure 2.2: The idea of MAB as slot machine play

In MAB problem, each action has an expected or mean reward that we receive upon selecting it, we refer to it as the *value* of the action. If we already had knowledge about all values of the actions, then solving the MAB problem is unnecessary as we would always select the action with the highest value. Hence in MAB, we assume that we do not know about the true value of each action or the dynamic of the environment.

At any time step, there is always at least one best arm decided by the estimated values of the actions over the plays, we called this a *greedy* action. One might intuitively want to play this action as *exploiting* the agent's knowledge about the environment. On the other hand, the other actions could be played, or *explored*, to improve the estimates of the nongreedy actions' value. Exploitation is the way to get the immediate maximum reward at the current step but exploration may produce a better result over the long run as we keep exploring, our estimate gradually becomes more correct to the true dynamic of the environment, as a result of the law of large number theorem.

Compared to the full RL problem, MAB has been simplified by omitting the situation, or the state, of the environment. However, it does not make MAB either an easily solvable problem or uninteresting. An extensive amount of effort has been put over the last decades into the many forms of MAB, theoretically or experimentally.

2.2.2 Action-value functions

Maintaining an estimate for each action's value, as previously mentioned, is a simple way of presenting the knowledge of the agent about the environment and is adopted in the idea of many MAB methods. We denote the true value of an arm k as $q(k)$ and the estimated value on the tth time step as $Q_t(k)$, so-called the action-value function. Recall that the true value of an arm is the expected reward given by selecting the arm. One common way to define the action-value function $Q_t(k)$ is to take the average of the rewards received by pulling arm k. Given that prior to time step t, arm k has been chosen $N_t(k)$ times and the trajectory of rewards for those selections is $R_1, R_2, \dots, R_{N_t(k)}$, then the action-value function is

$$Q_t(k) = \frac{1}{N_t(k)} \sum_{t=1}^{N_t(k)} R_t \tag{2.1}$$

If $N_t(k) = 0$, $Q_t(k)$ is usually initialised with a given value or one can uniformly play the arms once to avoid this case. As $N_t(k) \to \infty$, $Q_t(k)$ converges to the true value $q(k)$ of the arm following the law of large numbers. This is called the *sample-average* method to estimate the action values because each estimate is just an average of the reward samples. This is one simple way to define the action-value function and is good enough for many cases.

However, this formulation is not effective in certain situations. For example, if the environment is changing and the true values of the actions are prone to variation over time, this is referred to as a non-stationary MAB problem. In such cases, the rewards received from recent plays should be considered higher than the ones returned from the early time step. One of the many ways to track this non-stationary problem is using a discounted factor $\gamma \in (0, 1]$ in updating the estimate of an action whenever it is played as

$$Q_{t+1}(k) = Q_t(k) + \gamma(R_t - Q_t(k)) = (1-\gamma)^k Q_1 + \sum_{t=1}^{N_k(t)} \gamma(1-\gamma)^{N_k(t)-t} R_t \tag{2.2}$$

As the weight for each reward decays exponentially according to the exponent on $1-\gamma$, this method is called an exponential average.

Using the action-value function, at each time step, we can define the greedy action A_t^* as the one with the highest estimated action value:

$$A_t^* = \arg\max_k Q_t(k) \tag{2.3}$$

where $\arg\max_k$ denotes the value of k that produces the highest result for the subsequent expression (in case of a tie, the selection is made arbitrarily).

2.2.3 Regret

In RL as well as in MAB, the ultimate goal of the agent is to maximise the cumulative reward received over time hence total reward is the indicator of the performance of the agent or the strategy it is playing. Besides that, one standard approach is to compare the agent's cumulative reward to the *best-arm benchmark* as the possible total expected reward for a specific problem by always selecting the optimal arm. Consider the stationary MAB, define the optimal arm being k^* and recall that the true value of an arm is declared as $q(k)$. We can formally define the following quantity, called *regret* at round T as:

$$R(T) = q(k^*) \cdot T - \sum_{t=1}^{T} q(a_t) \tag{2.4}$$

with a_t as the arm selected at round t.

As indicated by its name, regret is a measure of how much reward the agent has lost by playing a suboptimal arm instead of the optimal one or how much it 'regrets' by not knowing the best arm in advance. Therefore, the agent's goal in MAB problems can also be considered as minimising the total regret over all time steps. While the reward is one standard benchmark for performance assessment in experiments, regret is a critical metric that is mostly used to evaluate the algorithms theoretically.

Chapter 3

Stationary Multi-armed Bandit with Bernoulli Reward

In the preceding chapter, we provided an overview of the fundamental principles and key concepts of Reinforcement Learning and Multi-armed bandit. In the upcoming section, we will plunge into a specific problem of Multi-armed Bandit, which involves independent and identically distributed Bernoulli rewards for each arm. Our approach begins by presenting a formal statement of the problem at hand, introducing the first research question, which afterwards is answered by an examination of two standard algorithms used to address it, *ε-greedy* and *Thompson Sampling* and a proposal of a new solution to the MAB problem, called *Surprised-based MAB method.* Finally, we present our second research question, the answer to which will be explored in the subsequent chapters of this report.

3.1 MAB model formulation

In this project, we consider a *finite-horizon* MAB model with *independent and identically distributed* (IID) rewards, so-called *stochastic bandits.* The agent has a set of K actions (or arms), denoted as $\mathcal{A} = \{1, 2, \dots, K\}$, to choose from and there is a *time horizon* T rounds, which is the fixed number of rounds (or time steps) for the game. At each time step $t \in \{1, 2, 3, \dots, T\}$, the agent chooses an arm a_t from $\mathcal{A}$ to play and receives a reward r_t sampled from a reward distribution $\mathcal{D}_{a_t}$ according to the selected arm. The ultimate goal of the agent is to maximise the cumulative reward received over the time horizon T, formally defined as $\sum_{t=1}^{T} r_t$. Our MAB problem formulation contains three key properties:

- **Stationary bandit:** The environment's characteristics are unchanged over the time horizon. This means that all the arms are available and their reward distributions are fixed for every time step.
- **IID Bernoulli rewards:** For every arm a, its corresponding reward distribution $\mathcal{D}_a$ is the Bernoulli distribution with mean $\mathbb{E}[\mathcal{D}_a]$. Every time an arm is chosen, the reward is sampled independently from this distribution. Hence, pulling an arm a would either yield a reward of 1 with probability of $\mathbb{E}[\mathcal{D}_a]$ or 1 with probability of $1 - \mathbb{E}[\mathcal{D}_a]$.

- **Partial observation:** At each round, the agent can observe nothing else but the reward for the selected arm. In particular, it can not obtain rewards for the other actions that are not selected.

Therefore, an agent interacts with the proposed MAB environment according to the protocol summarized as follows.

Problem protocol: Stationary MAB with Bernoulli Reward

Parameters: a set of K arms $\mathcal{A}$, time horizon of T rounds (both known to the agent); reward distributions $\mathcal{D}_k$ for $k = 1, 2, \ldots, K$ (unknown to the agent).
In each time step $t = 1, 2, \ldots, T$:

1. The agent selects one arm a_t.
2. A reward $r_t \in [0, 1]$ is sampled independently from $\mathcal{D}_{a_t}$.
3. The agent receives reward r_t.

Our MAB model presents a basic formulation of the MAB problem but is highly applicable in numerous scenarios as:

1. **Recommendation system:** In a recommendation system, each item represents an action of the bandit and the agent at each time, as a time step, selects an arm to display to the user. The system observes whether or not the user is interested in the item (by viewing or making a purchase), which corresponds to a reward of 1 or 0. It can be assumed that the probability of an item being engaged (mean reward) does not change over time, or at least for a sufficient period of time. The goal of the system is to gather as many engagements (cumulative rewards) from the user over a period of time.

2. **Medical trials:** A doctor who is treating patients can have several possible treatments, which are considered as the set of arms. The effectiveness of the treatment could be quantified as 1 for positive and 0 for not and be considered as the reward. Similar to the MAB agent, the doctor faces the problem of finding the arm that most likely produces a positive outcome.

Remark 3.1. Although our formulation only considers binary rewards, the methods we will discuss later in this report can be applied to MAB problems that have reward distributions with support of $[0, 1]$, i.e. the reward for selecting any arm is in $[0, 1]$.

Remark 3.2. Throughout this report, we use the following conventions. Arms or actions are denoted with a and time steps with t. There are K rounds and the time horizon is T rounds. The mean reward of an arm a is defined as $\mu_a \coloneqq \mathbb{E}[\mathcal{D}_a]$. The optimal arm (with the highest expected reward) is denoted as a^* with mean reward as $\mu^* = \max_{a \in \mathcal{A}} \mu_a$ and is not necessarily unique. The *gap* of an arm a, $\Delta_a \coloneqq \mu^* - \mu_a$, describes how bad the arm compared to a^*.

In the previous chapter, we discussed *regret* as a theoretical evaluation metric on how an algorithm performs in a MAB problem instance. Recall that regret presents how much the agent "regrets" not knowing the optimal arms in advance and is computed by comparing the agent's cumulative reward to the best-arm benchmark. Now, with the specified conventions, we can formally define the regret of the agent at round T as:

$$R(T) = \mu^* \cdot T - \sum_{t=1}^{T} \mu_{a_t} = \mu^* \cdot T - \sum_{a=1}^{K} \mu_a \cdot N_a \tag{3.1}$$

with N_a is the number of rounds that arm a is chosen.

Since N_a is a random quantity as it depends on the randomness in the rewards and/or in the agent's strategy, $R(T)$ is a random variable. Therefore, in theoretical analysis, we mainly care about the expected regret $\mathbb{E}[R(T)]$.

As introduced before, the exploration-exploitation dilemma is the representative challenge of RL and MAB. In our MAB model formulation, as a result of the *partial observation* property, at each round, the agent must choose to exploit its current knowledge or explore one of the other actions to update its understanding with the goal to find the real optimal arm. A MAB algorithm is a procedure guiding the agent on which behaviour it should choose from. In the remainder of this chapter, we will introduce three methods with different exploration-exploitation balancing strategies.

3.2 First research question

Having discussed the background, and formalising our problem, we now repeat the first research question of our project:

RQ1) What are the potential solution candidates to handle the exploration-exploitation trade-off in the MAB problem with Bernoulli reward signal?

Our answer to this question determines the algorithms that will be explored in this report. In the following sections, we will introduce two well-known MAB approaches, ε-greedy and Thompson Sampling, and propose a novel solution called Surprise-based Exploration policy. The candidates have different exploration strategies, and we aim to examine their efficiency in achieving high cumulative reward in our proposed MAB model.

3.3 ε-greedy

3.3.1 Algorithm description

ε-greedy is a simple approach in RL and MAB, first described by Watkins [23], which handles exploration-exploitation trade-off by behaving greedily most of the time, but occasionally instead to randomly choose among all of the available actions with equal probability to explore potentially better options. Specifically, at each round, the algorithm selects the action with the highest action-value estimate with probability of $1 - \varepsilon$ and selects a random action with probability of ε (see Algorithm 3.1).

Algorithm 3.1: ε-greedy

```
Input parameters: $\varepsilon \in (0, 1)$
for time step $t = 1, 2, \ldots, T$ do
    $u \leftarrow$ uniform random number in $(0, 1)$
    if $u < \varepsilon$ then
        explore: randomly choose an arm from the arm set.
    else
        exploit: choose the arm with the highest action-value estimate.
    end if
end for
```

In this algorithm, the best arm is decided by the action-value estimate of each arm, that the arm with the highest value is determined to be the current optimal one. Since our MAB model is a *stationary bandit*, the action-value function in Equation 2.1 is used by the algorithm, in which the value of each arm is the unweighted mean rewards received by playing that arm.

The hyper-parameter ε, which is manually decided in range $(0, 1)$, determines the degree of exploration versus exploitation, with a high value of ε encouraging exploration and a low value of ε encouraging exploitation. Therefore, the choice of ε has a great impact on the performance of the algorithm. In the beginning, the agent might benefit from the exploration-promoted policy of a high ε and quickly gain knowledge about the environment, which, in this case, is the true mean reward of the arms. But after having a good estimate of the environmental characteristics, the agent might want to change to an exploitation-oriented policy with a low ε for a higher reward received. Inspired by this idea, some adaptive ε-greedy algorithms were introduced, in which ε decreases over time. Theoretically, Cesa-Bianchi and Fisher[3] achieved a lower expected regret bound for this variant of the algorithm compared to the canonical constant ε algorithm. However, in an empirical study, Vermorel and Mohri[21] did not find any practical advantage to using the method. Therefore, in our project, we only consider fixed values of ε, as presented in Algorithm 3.1.

3.3.2 Theoretical upper regret bound

As stated, the *expected regret* is the main focus in MAB algorithm theoretical analysis. Firstly, we start with the following Lemma.

Lemma 3.1. *Given a MAB environment as formulated with K arms, at time step t, every arm is chosen n times. The probability of a suboptimal arm being chosen in an exploitation round is bounded as*

$$\mathbb{P}(\exists_k \, \hat{\mu_k} \geq \hat{\mu_*}) \leq \min\left(1, \sum_k \exp\left(-\frac{n\Delta_k^2}{2}\right)\right)$$

Proof. Refer to Appendix A.3. □

This lemma shows that the maximum probability of optimal arm misidentification could be higher if there are more arms and/or if the mean reward gaps of the arms is smaller and decrease over time where n increases, where the agent gathers more information about the environment.

Now, we provide an upper bound for ε-greedy $\mathbb{E}[R(t)]$ for every time step t as follows.

Theorem 3.2. *For the Multi-armed bandit problem with Bernoulli reward with K arms, ε-greedy with exploration term ε achieves the upper regret bound*

$$\mathbb{E}[R(t)] < \frac{\varepsilon}{K} \sum_{a \in \mathcal{A}} \Delta_a + (1-\varepsilon) \cdot \min \left(1, \sum_{a \in \mathcal{A}} \exp\left(-\frac{\varepsilon t \Delta_a^2}{2K}\right)\right) \cdot 2r$$

for every time step t with $r = \sqrt{\frac{2K \log t}{\varepsilon t}}$.

Proof. Refer to Appendix A.4. □

Theorem 3.2 indicates that the regret upper bound for each round t, which is equivalent to the cumulative reward lower bound, is higher if we increase the number of arms and/or the suboptimal arms have the mean rewards being closer to the mean reward of the optimal arm. Also, this regret bound also decreases over the time horizon.

3.4 Thompson Sampling

3.4.1 Bayesian Learning for Bernoulli distribution

Consider the problem of learning parameter(s) of a parametric distribution from observed data, for example, learning the parameter μ of a Bernoulli distribution, with p as the probability of 1 and $1-p$ as the probability of 0. The gathered data sampled from the distribution as

$$0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0$$

As there are 6 of 1s and 6 of 0s in the data, a frequentist's approach would estimate the value of μ as 0.5 (six of 1s and six of 0s) with some confidence.

On the other hand, a Bayesian learner adopts a different approach by maintaining a probability distribution over μ to present its uncertainty about the parameter. The probability distribution represents the likelihood of the parameter μ being a specific value. Before having the data, this distribution is called the *prior*, which encodes the learner's initial belief about the parameter. After observing the data, the learner adjusts its belief using Bayes's rule, resulting in an updated *posterior* distribution.

Getting back to the mentioned example of learning the parameter μ of a Bernoulli distribution. The Bayesian learner starts with a prior $p(x)$ that:

$$p(x) = \mathbb{P}[\mu = x]$$

After observing the sampled data D with 12 elements as above, the learning obtains a posterior, using the Bayes rule:

$$\mathbb{P}[\mu = x | D] = \frac{\mathbb{P}[D | \mu = x] \cdot \mathbb{P}[\mu = x]}{\mathbb{P}[D]} \propto \mathbb{P}[D | \mu = x] \cdot p(x)$$

with $\mathbb{P}[D|\mu = x]$ is the probability of generating data D from the Bernoulli distribution with parameter x.

One problem with Bayesian learning is the normalisation term $\mathbb{P}[D]$ is usually intractable or overly complicated. However, sometimes there are closed-form solutions for the posterior distribution given the prior and the observations. In particular, for Bernoulli samples, Beta distribution is a conjugate prior for the Bernoulli distribution, i.e. we can choose the prior as a Beta distribution and the posterior would also be a Beta distribution.

Definition 3.1. A *Beta distribution* with two parameters, (α, β) is a probability distribution having support $(0, 1)$ and a probability density as

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1 - x)^{\beta-1}$$

where $\Gamma(x)$ is the Gamma function. For $x \in \mathbb{Z}^+, \Gamma(x) = (x - 1)!$.

Remark 3.3. Given a Bernoulli parameter learning problem and a prior as a Beta distribution with parameters (α, β). We observe a sample $r \in \{0, 1\}$, the posterior is a Beta distribution with parameter $(\alpha + r, \beta + 1 - r)$.

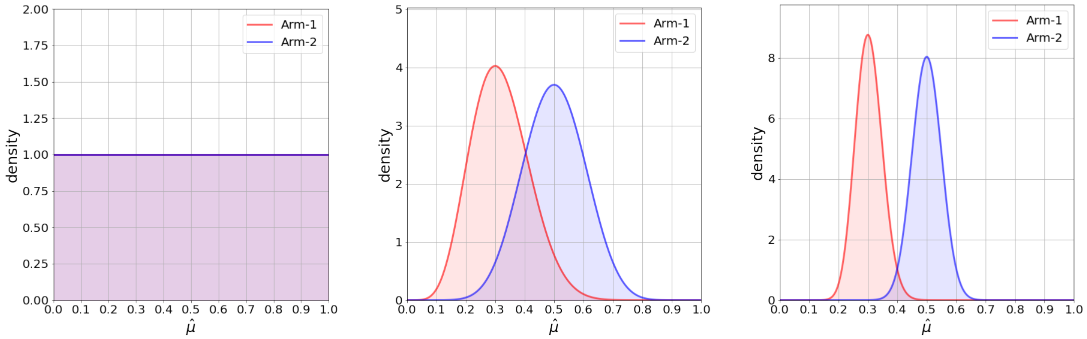
3.4.2 Bernoulli Thompson Sampling

Thompson Sampling, first proposed by William R. Thompson[20], is one of the oldest heuristics in solving MAB problems. However, it was mostly ignored until recent empirical studies [15, 4] showing its great performance then the algorithm has been widely researched in the academia[1, 13] and adopted in the industry [5, 19]. The core idea of Thomson Sampling is to use Bayesian inference to learn the reward distribution of each arm. In particular with our MAB formulation with Bernoulli reward, the actual mean reward μ_a of an arm a is modelled using Beta distribution with the parameters updated every time a reward is received by playing that arm using Remark 3.3. At every time step, the agent samples from the prior distributions of the arms and the one with the highest sampled value is chosen. The algorithm is called Bernoulli Thomson Sampling and formalised as in Algorithm 3.2.

Algorithm 3.2: Bernoulli Thompson Sampling

```
Initialisation: $S_a = 0, F_a = 0$ as the number of successes (reward = 1)
 and failure (reward = 0) of an arm $a$ for $a \in \{1, 2, \dots, K\}$
for time step $t = 1, 2, \dots, T$ do
    $\hat{\mu}_a \sim Beta(S_a + 1, F_a + 1)$ for $a \in \{1, 2, \dots, K\}$
    Chosen arm $a_t \leftarrow \max_a \hat{\mu}_a$
    if $r_1 = 1$ then
        $S_{a_t} \leftarrow S_{a_t} + 1$
    else
        $F_{a_t} \leftarrow F_{a_t} + 1$
    end if
end for
```

To understand how Thompson Sampling handles the exploration-exploitation dilemma, we look at Figure 3.1, where we plot the density functions of the Beta distributions as priors of two arms with true mean rewards of 0.3(Arm-1) and 0.5(Arm-2), in different numbers of pulls $n = 0, 20, 100$ for both arms. Here we assume that the number of observed successes (reward $= 1$) follows exactly their true expected rewards, i.e. $0.3 \times n$ for Arm-1 and $0.5 \times n$ for Arm-2.



(a) $n = 0$ (b) $n = 20$ (c) $n = 100$

Figure 3.1: The density functions of two beta distributions modelling mean rewards of two arms, in different number of pulls

Initially, when none of the arms is pulled ($n = 0$), the prior distributions, $Beta(1, 1)$, take the form of a Uniform distribution over $[0, 1]$. As a result, when sampling from the two priors, the probabilities of each arm being selected are the same. This represents the highest level of exploration when the algorithm has no information about the environment. As the number of trials n increases, the priors' densities become more concentrated around the true mean rewards, causing the chance of Arm-1 being selected over Arm-2 to diminish over time. In these cases, the algorithm obtains more knowledge about the environment and has higher confidence about the estimated mean reward of the arms, resulting in a greater priority on exploiting Arm-2.

Compared to ε-greedy, Thompson Sampling has two major differences. Firstly, ε-greedy uses the frequentist's method to estimate the mean rewards of the arms, i.e. a plug-in estimator based on the trajectory of rewards, while Thompson Sampling uses Bayesian inference to learn the reward distributions. Secondly, ε-greedy explicitly defines the degree of exploration as the value of ε. On the other hand, Thompson Sampling implicitly handles exploration and exploitation over the differences in priors of the arm rewards.

3.4.3 Theoretical upper regret bound

Recently, Thompson Sampling has attracted notable attention from scientists. Besides empirical studies (e.g. [12] by May and [4] by Chapelle and Li) that demonstrated the efficiency of Thompson Sampling in experiments, there also exists theoretical analysis on the method, such as [6] by Granmo, [9] by Kaufmann and [1] by Shipra and Navin. Here, we refer to the work of Shipra and Navin [1] with the tightest upper bound on the expected regret for the MAB problem with Bernoulli reward as follows.

Theorem 3.3. *(**Theorem 2** in [1]). For the K-armed stochastic bandit problem with Bernoulli reward, Thompson Sampling has the expected regret*

$$\mathbb{E}[R(T)] \leq \mathcal{O}\left(\left(\sum_{a\in\mathcal{A}} \frac{1}{\Delta_a}\right)^2 \ln T\right)$$

in time T, where Δ_a is the gap of arm a (Remark 3.2).

This theoretical regret upper bound shows that the agent will probably have a lower cumulative reward if there are more arms and the arms are more competitive, i.e. the mean reward gaps of the arms are lower.

From Theorem 3.2 and Theorem 3.3, we can summarise that the hardness of the Multi-armed bandit model with Bernoulli reward signal is defined by two terms: (1) the number of arms K, higher K means harder problem and (2) the competitiveness of the arms, that if the suboptimal arms have closer mean rewards to the true optimal arm, it is more difficult for the agent to achieve the maximum potential cumulative reward.

3.5 Surprise-based Exploration Policy

In psychology, the motivation to do a specific action is classified into two categories, *extrinsic motivation*, when we are motivated to perform a behaviour because of some rewarding outcome and *intrinsic motivation*, which involves being motivated to do something because it is inherently pleasurable. In the context of RL, most of the methods focus solely on the former one, which uses external rewards given by interacting with the environment as the only motivation. However, some studies (e.g. [16]) have proposed that intrinsic motivation can be used to encourage exploration and learning by providing an internal reward signal that is independent of the external rewards provided by the environment to help the agent improve its overall performance.

In this section, we will study how to incorporate the idea of intrinsic motivation into MAB problem by first discussing a previous work of Houthooft *et al.*, which introduced an additional intrinsic reward to the reward received by the Markov Decision process environment in RL. Inspired by this idea, we later propose a similar exploration-promoting solution for our MAB problem.

3.5.1 Related Work

In this part, we review the work *'VIME: Variational Information Maximizing Exploration'*[7] by Houthooft *et al.*. They introduced a curiosity-driven exploration strategy for a Markov Decision Process (MDP) framework in RL, based on the maximisation of information gained about the agent's belief of the environment based on the Kullback-Liebler divergence, as defined below. Within their method, the agent is encouraged to explore state-action regions that are relatively unexplored.

Definition 3.2. For two probaility distribution P and Q over the same sample space $\mathcal{X}$, the **Kullback–Leibler divergence** (or **KL-divergence**) from P to Q is defined as:

$$D_{KL}(P||Q) = \begin{cases} \displaystyle\sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} & \text{if } P, Q \text{ are discrete} \\ \\ \displaystyle\int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)} dx & \text{if } P, Q \text{ are continuous} \end{cases}$$

The KL-divergence is a measure of how distribution P is different from distribution Q.

In Houthooft *et al.*'s work, the agent models the environment dynamics via a model $p(s_{t+1}|s_t, a_t; \theta)$, parameterised by the random variable Θ with values $\theta \in \mathbf{\Theta}$, where s_t and a_t are respectively the state and action at time step t. The agent's initial knowledge about Θ is summarized by a prior density $p(\theta)$, which is updated in a Bayesian manner. The trajectory of the agent up until time step t is denoted as $\xi_t = \{s_1, a_1, \ldots, s_t\}$. The *information gain* is defined as the KL-divergence from the agent's new belief over the dynamics model to the old one after choosing action a_t and moving to new state s_{t+1}, formally as:

$$S = D_{KL}(p(\theta|\xi_t, a_t, s_{t+1})||p(\theta|\xi_t))$$

The agent is encouraged to explore the environments by adding the term S as an *intrinsic reward*, which captures the agent's surprise in the form of a reward function, to the external reward. The new reward that the agent receives now is as follows:

$$r'(s_t, a_t, s_{t+1}) = r(s_t, a_t) + \eta D_{KL}(p(\theta|\xi_t, a_t, s_{t+1})||p(\theta|\xi_t))$$

with $\eta \in \mathbb{R}_+$ as a hyper-parameter controlling the urge to explore.

3.5.2 Proposed surprise-based exploration for Multi-armed Bandit with Bernoulli Reward

Inspired by the work of Houthooft *et al.*, here we introduce a surprise-based exploration strategy for the stochastic MAB problem with Bernoulli reward adopting the idea of integrating a surprise term as an intrinsic reward to the total reward that the agent receives in each time step.

In our proposed method, the agent's belief about the environment is modelled as a set of probability distributions $p(\mu_a)$ for $a \in \mathcal{A}$, which presents the agent's knowledge about the mean reward of an arm a. Up until step t, given the trajectory of external reward that the agent receives from interacting with the environment as $\xi_t = \{a_1, r_1, a_2, r_2, \ldots, a_{t-1}, r_{t-1}\}$. If an arm a_t is selected and returns a reward r_t, the *surprise term* is formalised as the difference between the new belief and old belief about arm a_t as follows:

$$S(\xi_t, a_t, r_t) = D_{KL}(p(\mu_{a_t}|\xi_t, r_t)||p(\mu_{a_t}|\xi_t))$$

In words, the surprise term $S(\xi_t, a_t, r_t)$ is the KL-divergence from the distribution of the mean reward of arm a_t after being pulled to the distribution before being pulled.

Similar to what is done in VIME, the agent, instead of receiving the reward r_t, obtains a new reward r'_t with the additional surprise term as an intrinsic reward:

$$r'_t = r_t + \eta S(\xi_t, a_t, r_t)$$

with $\eta \in \mathbb{R}_+$ as a parameter controlling the urge to explore.

Remark 3.4. The agent keeps track of both rewards r_t and r'_t at every round. r_t is used to update the distribution $p(\mu_{a_t})$ following Bayesian inference while r'_t is used in the arm selection process.

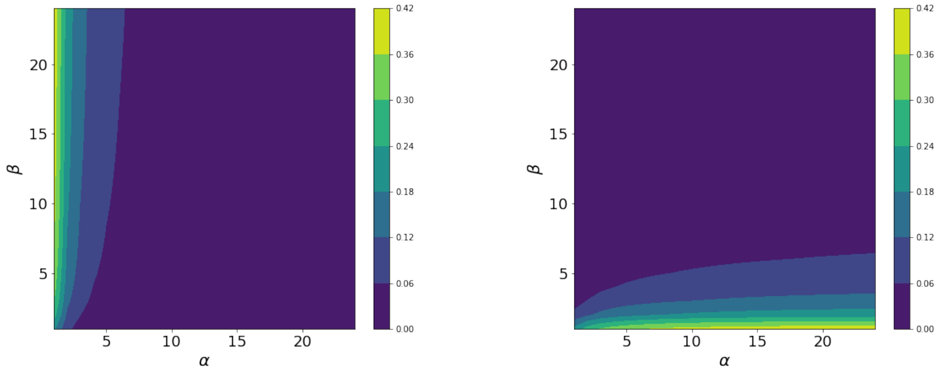
Similar to what we formalised in Thompson Sampling, the distribution over the mean reward μ_a of an arm a is modelled using a Beta distribution since we have a closed-form solution for the posterior, which is also a Beta distribution, after observing new reward after pulling a. In particular, given that at time step t, the chosen arm a_t has S_a and F_a as the numbers of successes (reward = 1) and failures (reward = 0) before being pulled, then the prior $p(\mu_{a_t}|\xi_t))$ is modelled as $Beta(S_a + 1, F_a + 1)$. The posterior $p(\mu_{a_t}|\xi_t, r_t)$ is $Beta(S_a + 1 + r_t, F_a + 1 + 1 - r_t)$ with $r_t \in \{0, 1\}$ as the reward received by choosing a_t. The surprise term $S(\xi_t, a_t, r_t)$ is calculated using the following preposition.

Proposition 3.1. *Let $Q \sim Beta(\alpha, \beta)$, $P \sim Beta(\alpha + r, \beta + 1 - r)$ with $\alpha, \beta \in \mathbf{Z}_+, r \in \{0, 1\}$. The KL-divergence from P to Q is calculated as*

$$D_{KL}(P||Q) = \begin{cases} \ln \dfrac{\alpha+\beta}{\alpha+1} + \psi(\alpha+1) - \psi(\alpha+\beta+1) & \textit{if r=1} \\ \\ \ln \dfrac{\alpha+\beta}{\beta+1} + \psi(\beta+1) - \psi(\alpha+\beta+1) & \textit{if r=0} \end{cases}$$

where $\psi(x) = \frac{\Gamma(x)'}{\Gamma(x)}$ is the Digamma function and $\Gamma(x)$ is the Gamma function.

In Figure 3.2, we visualise the value of the KL-divergence in Proposition 3.1, which is equivalent to the value of the surprise term $S(\xi_t, a_t, r_t)$, with different values of α and β, for two cases, $r = 1$ and $r = 0$. Note that α and β correspond to the number of successes and failures added by 1 for arm a_t.



(a) $r = 1$

(b) $r = 0$

Figure 3.2: The values of the KL-divergence in Proposition 3.1 with different values of α and β for $r = 1$ and $r = 0$

Looking at the visualisation, we can see that if the larger β being compared to α, as the agent previously observed more failures than successes from the selected arm, the KL-divergence, or the surprise term, is bigger if the returned reward $r_t = 1$, which means the agent is more surprised if it gets a success from pulling the arm at the current time step. On the other hand, if $r_t = 0$, the agent is barely surprised as the KL-divergence is close to 0.

3.5.3 Surprise-based ε-greedy

The proposed surprise-based exploration policy is alone not a complete solution to the exploration-exploitation dilemma in MAB but rather a way to promote exploration by adding an intrinsic reward as the surprise term to the environment's reward that the agent receives. This method is a way to encourage the agent to visit less visited arms in order to gain more knowledge about the environment and hence get more confidence about its detection of the optimal arm. The new reward, which is the combination of a surprise-based intrinsic reward and the external reward, can be integrated into MAB algorithms to elevate exploration in the early stage of the game. As an example, here we propose the alternative algorithm of ε-greedy incorporating the idea, so-called *Surprise-base ε-greedy*, as described in Algorithm 3.3.

Algorithm 3.3: Surprise-based ε-greedy

```
Input parameters: $\varepsilon \in (0,1), \eta \in \mathbf{R}_+$
Initialisation: $S_a = 0, F_a = 0$ as the number of successes (reward = 1)
  and failure (reward = 0) of an arm $a$; $R_a, N_a$ as the mean reward, and
  number of pulls of arm $a$, for $a \in \{1, 2, \dots, K\}$;
for time step $t = 1, 2, \dots, T$ do
    $u \leftarrow$ uniform random number in $(0,1)$
    if $u < \varepsilon$ then
        explore: randomly choose an arm from the arm set.
    else
        exploit: chosen arm $a_t \leftarrow \max_a R_a$
    end if
    $r_t \leftarrow$ reward returned by selecting $a_t$
    $r'_t = r_t + \eta D_{KL}(Beta(S_{a_t} + 1 + r_t, F_{a_t} + 2 - r_t)||Beta(S_{a_t} + 1, F_{a_t} + 1)$
    $R_a \leftarrow R_a + \frac{r'_t - R_a}{N_a + 1}$
    $S_a \leftarrow S_a + r_t$
    $F_a \leftarrow F_a + 1 - r_t$
    $N_a \leftarrow N_a + 1$
end for
```

In Surprise-based ε-greedy, besides the explicit declaration of the exploration rounds with probability of ε, by using the newly introduced rewards, in the exploitation rounds, the agent is more likely to select arms that are most misunderstood about their reward distributions. Compared to the original algorithm, we now add a level of exploration into the exploitation rounds, which is determined by the parameter η and decreases over time, as the agent collects more sufficient knowledge about the environment.

3.6 Second research question

After discussing the background, formalising our problem, and proposing candidate solutions, we now return to the main topic of our project as *'Exploitation-Exploration in Multi-armed Bandit with Bernoulli Reward'* by first investigating the following second research question in the remainder of this report.

RQ2) What is the empirical performance of the solution candidates in different scenarios of the MAB environment?

This question focuses on the central goal of the agent in the MAB problem, which is maximising the cumulative reward received over the time horizon. Additionally, we analyse how different algorithms perform in terms of discovering the best arm using their exploration-exploitation strategies. By evaluating our surprise-based ε-greedy algorithm based on these criteria, we aim to determine whether it enhances the performance of the original method. Finally, we discover the extreme cases, where the algorithms fail to deliver satisfactory outcomes. This prompts us to investigate a second research question for our project.

Chapter 4

Multi-armed Bandit Algorithms Evaluation

The previous chapter covered the formulation of our Multi-armed bandit model and the methods that were investigated in this project. In this chapter, we will be presenting the results of our experiments, which aim to answer the research question RQ2: "What is the empirical performance of the proposed algorithms in various scenarios of the MAB environment?". To begin with, we present our experiment procedures including experimental settings and evaluation criteria utilized in our research. Following this, we will provide the results obtained from the experiments. Finally, we will discuss the results and our findings, from which we will introduce a MAB problem of extreme cases.

4.1 Methodology

In this section, we will outline the settings and procedures that were employed in our experiments, beginning with the choices of MAB environment variables. To assess the efficacy of the proposed algorithms, we conducted an evaluation across three scenarios of the formulated MAB problem as follows.

- **4-armed bandit**: A MAB problem with 4 arms and the true mean rewards of them being $[0.4, 0.5, 0.6, 0.7]$.
- **10-armed bandit**: A MAB problem with 10 arms and the true mean rewards of them being $[0.3, 0.4, 0.4, 0.45, 0.5, 0.50, 0.55, 0.6, 0.65, 0.7]$. In the added arms, there is a competitive arm, with the mean reward of 0.65 to the optimal arm, which has a mean reward of 0.7.
- **40-armed bandit** An extreme case with numerous arms for the agent to choose from. Similar to *10-armed bandit*, we have other 4 competitive arms with mean rewards close to the optimal arm.

The level of difficulty in MAB problems is mostly determined by the number of arms and the mean reward distribution. In particular, more arms and/or more competitive optimal arms require the agent to have more exploration to acquire knowledge about the environment and find the optimal arm. By varying the two variables, we can comprehensively evaluate the algorithms' ability to tackle different cases of MAB problems.

For all three scenarios, we set the time horizon T to be 1000 rounds. Since there are randomness in the reward signal and the algorithms, we run every game instance, which includes one MAB scenario and one algorithm, with 1000 repetitions. This helps us see the variability of the algorithms' performance.

In our experiments, to see how well the algorithms did on the MAB instances, we recorded and analysed the following criteria

- **Cumulative reward**: This is the primary goal of the agent in RL and MAB. The algorithm with the ability to achieve higher reward over time is typically deemed to be more effective.

- **Percentage of optimal choices:** To acquire the best reward possible, the agent should be able to find the best arm to exploit in the long run. This criterion shows us the effectiveness of the algorithm in exploring and gathering information about the environment.

Now that we have provided an overview of our experimental procedures, the subsequent section will present the outcomes of our experiments.

4.2 Results

4.2.1 4-armed bandit

Firstly, Figure 4.1 presents the distribution of the rewards at the termination point, i.e. after 1000 rounds, with 1000 repetitions. The blue dashed line stands for the best-arm benchmark, which is the expected cumulative reward if the optimal arm is selected at every round.

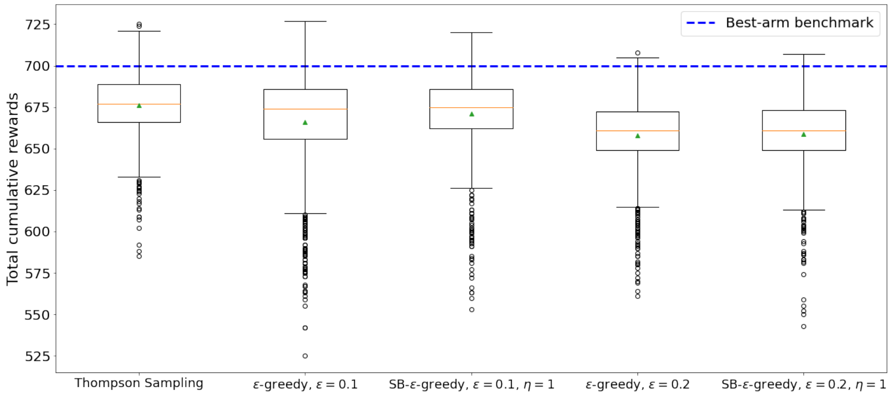


Figure 4.1: The cumulative rewards after 1000 rounds with 1000 repetitions. *SB-ε-greedy stands for the proposed Surprise-based $\varepsilon - greedy$ algorithm.*

It can be observed that overall, the algorithms averagely achieve more than 90 per cent of the best-arm benchmark while Thompson Sampling obtains the highest mean reward, tightly winning over Surprised-based *epsilon*-greedy with parameters $\varepsilon = 0.1, \eta =$

1 at the second place. The effectiveness of the additional surprised-based reward makes insignificant improvement for the ε-greedy algorithm in both cases of the experimented ε parameter.

In Figure 4.2, we present the percentage of optimal arms chosen along the rounds.

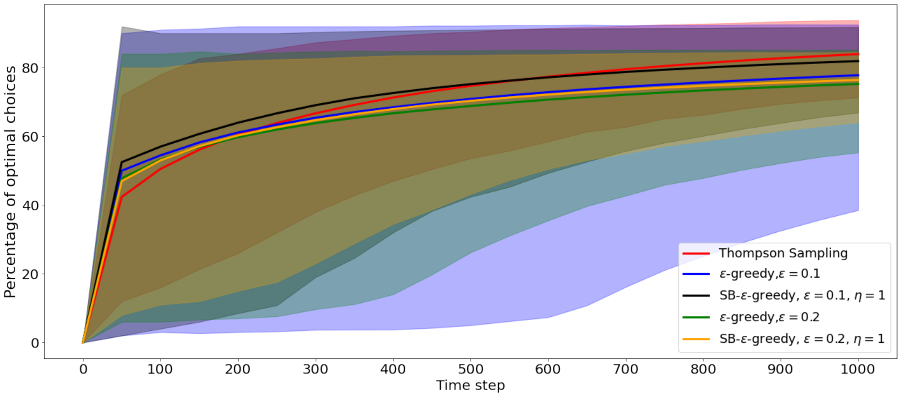


Figure 4.2: The percentage of optimal choices over time. *SB-ε-greedy stands for the proposed Surprise-based $\varepsilon - greedy$ algorithm. The lines and corresponding coloured-fill areas correspond to the mean value and 80%-confidence interval for each strategy.*

On average, the algorithms do mostly exploration in the first 50 rounds of the game and eventually pull the actual optimal arm for $70 - 80$ per cent rounds at the termination point, with Thompson Sampling at the first place, which correlates to its performance in the cumulative reward. However, the variation of their performance is quite large, especially ε-greedy with $\varepsilon = 0.1$.

4.2.2 10-armed bandit

Figure 4.3 visualises the total reward achieved by the strategies after 1000 rounds.

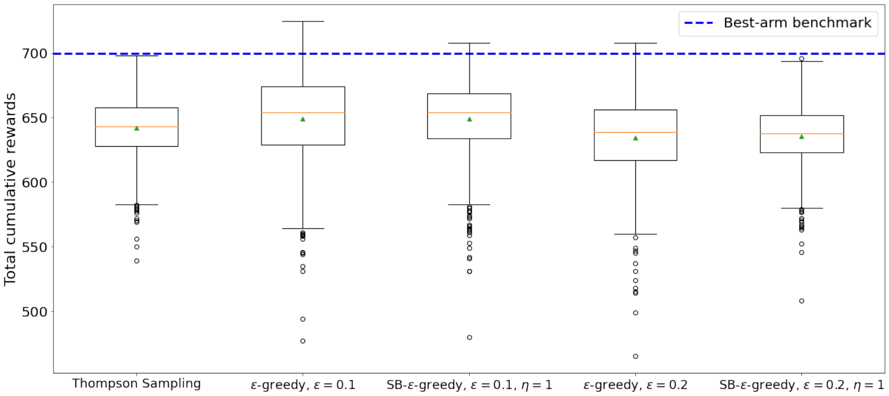


Figure 4.3: The cumulative rewards after 1000 rounds with 1000 repetitions. *SB-ε-greedy stands for the proposed Surprise-based $\varepsilon - greedy$ algorithm.*

As expected, the increment in the number of arms and an additional close-optimal arm causes performance decrement in all algorithms with Thompson Sampling being affected the most.

Figure 4.4 shows similar results as the algorithms can only pull the optimal arm for about 50 per cent of the time steps. The variation of the performance on this criterion is even larger in this instance, compared to the 4-armed bandit.

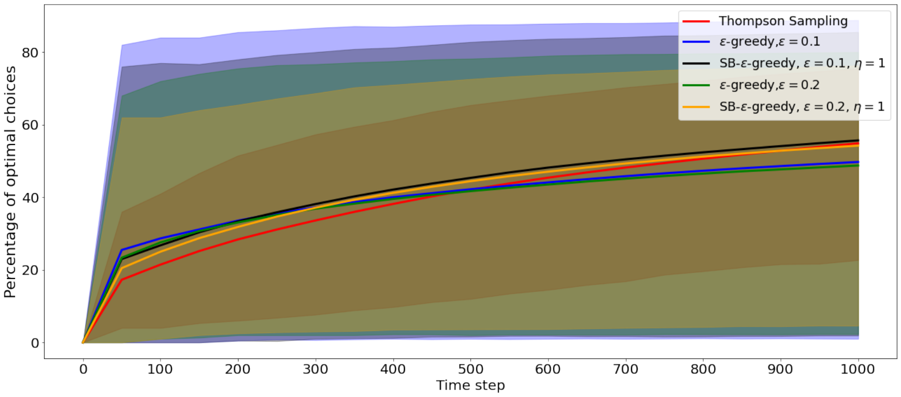


Figure 4.4: The percentage of optimal choices over time. *SB-ε-greedy stands for the proposed Surprise-based $\varepsilon -$ greedy algorithm. The lines and corresponding coloured-fill areas correspond to the mean value and 80%-confidence interval for each strategy.*

4.2.3 40-armed bandit

In our experiments, we also evaluate the performance of the algorithms in an extreme case with 40 arms and more competitive options to the optimal arm. In Figure 4.5 and 4.6, we present the results from the experiment with this scenario in terms of cumulative reward and percentage of optimal choices over time, respectively.

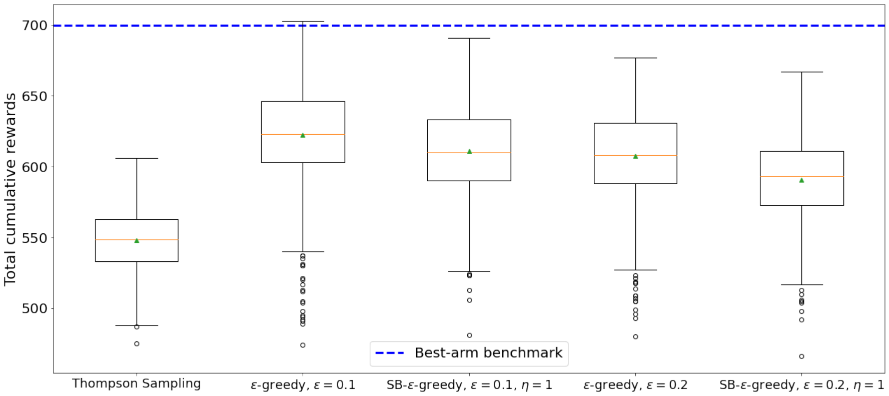


Figure 4.5: The cumulative rewards after 1000 rounds with 1000 repetitions. *SB-ε-greedy stands for the proposed Surprise-based $\varepsilon -$ greedy algorithm.*

As we once again expand the action set for the agent, the algorithms perform more poorly and Thompson Sampling now becomes the worst strategy of all methods. Another change here is that our surprise-based exploration policy makes ε-greedy achieve a

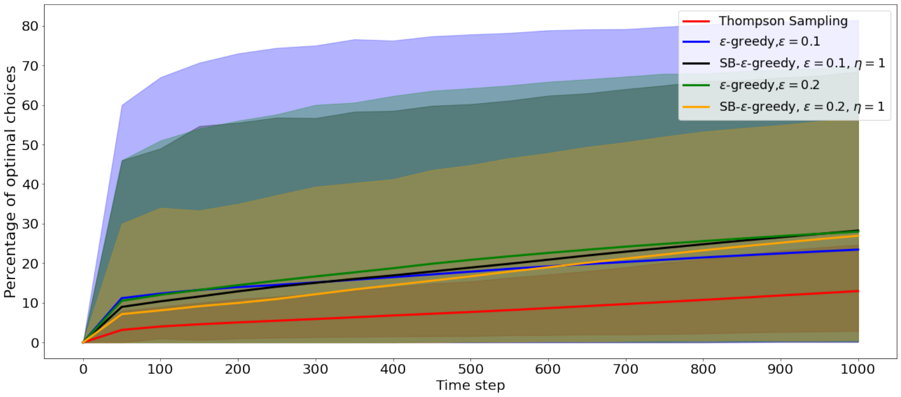


Figure 4.6: The percentage of optimal choices over time. *SB-ε-greedy stands for the proposed Surprise-based $\varepsilon - greedy$ algorithm. The lines and corresponding coloured-fill areas correspond to the mean value and 80%-confidence interval for each strategy.*

lower average cumulative reward in both values of the parameter ε. In terms of optimal choice percentage, Thompson Sampling reaches a much lower value, at about 10 per cent, compared to other methods with around 20 per cent.

4.3 Discussion

In this section, we summarise and discuss the observed results from our experiments with the proposed MAB environment instances. Firstly with a relatively small number of arms, Thompson Sampling is clearly the best choice for handling the trade-off between exploration and exploitation in the problem and our proposed surprise-based policy with additional intrinsic reward can make improvements to the original ε-greedy algorithm, though not significantly. Furthermore, the exploration promotion in the policy produces another enhancement by making the cumulative rewards more stable and predictable over many repetitions. This effect can also be seen in the optimal choice percentage criterion as Surprise-based ε-greedy has notably lower variation compared to the ε-greedy algorithm.

However, when the arm set is enlarged and more competitive arms represent, Thompson Sampling suffers severely and becomes less effective in both terms of achieving high reward and finding the optimal arm. This can be explained through its arm selection mechanic. At any time step, a value sampled from a poor arm's reward distribution, which is modelled as a Beta distribution, always has a probability to be larger than the sample from the good arm, hence the arm is chosen over the actual better arm. As the number of arms increases, there is a higher probability of a suboptimal arm being selected, making it more difficult for Thomson Sampling to get a high reward. In particular, we compare Thompson Sampling and ε-greedy in the scenario where both algorithms already found the optimal arm (with the highest empirical reward) with some confidence. In ε-greedy, the probability for that empirically optimal arm being chosen is a constant of $1 - \varepsilon$ regardless of the number of arms. However, in Thompson Sampling, this probability decreases as the number of arms increases, resulting in a decline in performance. Additionally, with more arms, the mentioned reward improvement from the proposed surprise-based policy gets smaller and eventually turns into a negative effect, even though the Surprise-based

ε-greedy method still has a higher percentage of rounds that the optimal arm is pulled.

In the extreme case with 40 arms, all algorithms struggle to achieve the ultimate goal of maximising the reward. Compared to the best-arm benchmark, we see that there is still room for improvement, especially with Thompson Sampling. Moreover, the percentage of optimal choices increases over time, however, at a very slow pace and only averagely reaches at highest of 20 per cent with Surprise-based ε-greedy. With this observation, we deem that a large arm set poses another challenge to the MAB problem. In the next chapter, we will discuss this problem and propose a solution to solve the issue.

Chapter 5

Arm Reduction in Multi-armed bandit

The experimental results presented in the last chapter have highlighted the challenge of dealing with Multi-armed Bandit problems that involve a large number of arms. In this chapter, we will present our solution proposed to address the challenge. To begin, we will provide a concise problem statement that outlines the difficulties associated with MAB problems featuring a large number of arms. We will then present our novel solution to this problem, which involves an arm reduction algorithm. Lastly, we will introduce our third important research question.

5.1 Problem statement

Empirically, we can see from the experimental results in chapter 4 that all algorithms struggle to excel in a MAB environment with a large number of arms. Theoretically, the presented upper regret bounds for ε-greedy and Thompson Sampling in chapter 3 both depend on the number of arms K that increasing K would also increase the bounds for both algorithms.

The large number of arms raises several problems in the context of MAB. Firstly, it demands a considerable amount of time steps for an algorithm to gain sufficient knowledge about the environment and identify the good arms with the highest expected rewards. As the time horizon is finite, there are fewer time steps for exploiting the good arms, hence, the cumulative reward is lower. Additionally, due to a greater number of actions available, the probability of a suboptimal arm over-performing when selected, which may lead to incorrect optimal arm identification, is higher, resulting in suboptimal performance. Finally, over the long run, having more arms also means more chances for bad arms being chosen over good arms. For example, in an exploration round of ε (with probability of ε), as the arms are selected randomly with equal probability, the presence of many suboptimal arms reduces the chance of the optimal arms being pulled in the round.

Thus, finding effective strategies to address these problems becomes important for optimizing the performance of MAB algorithms. In the next section, we will introduce our proposed solution for overcoming the challenge of a large number of arms in MAB.

5.2 Proposed solution

5.2.1 General solution

To address the challenge, we propose a solution by splitting the time horizon into two distinct stages as follows.

- **Stage 1: Best arms identification** In this stage, we aim to narrow down the best m arms in the set of K arms.
- **Stage 2: Standard MAB** Given the result from stage 1, a regular MAB algorithm is employed for the rest of the time horizon, with the input as the remaining arms.

The first stage is the pure exploration phase, where we temporarily forget about the ultimate goal of maximising the received reward, instead, we try to gather efficient knowledge about the environment by employing an algorithm which pulls the arms with a specific strategy such that in the end of the stage, we can eliminate the poorest arms out of consideration with the high confidence. In the second stage, with the surviving arm from the first stage, the primary objective of attaining high cumulative reward is revisited and approached through a standard MAB algorithm.

By following this approach, a new dilemma arises, which is distributing time steps for each stage. A long pure exploration stage can increase our confidence in the results by ensuring that the actual best arms are detected and remain in the final output. However, this approach may come at the expense of the exploitation stage, where the goal is to maximize the reward. On the other hand, if the exploration stage is too short, we have more rounds to earn rewards in the second stage, but the likelihood of error in the result of the first stage is higher.

The same problem also occurs in the choice of m, the number of arms remaining in the set that will be 'exploited' in the second stage. A small m may improve the performance of the second stage as the standard MAB works better with a smaller number of arms as we discussed. However, this increases the chance of erroneous detection in the first stage. Conversely, if m is too large, we may achieve a higher level of confidence in the results of the best arm identification algorithm but challenges posed by a large arm set may still remain.

5.2.2 Successive-Reject Best Arms Identification

In this part, we will introduce our algorithm used in the first stage of the solution, which is to detect the subset of best arms. Firstly, we look at the problem of *Best Arm Identification* in MAB, which received considerable attention in the academic literature. The problem is generally categorised into two groups: **(i)** the *fixed confidence* setting, where the objective is to minimise the number of time steps required to find the best arm with a certain confidence, and **(ii)** the fixed-budget setting, where the agent attempts to maximise the probability of correct identification given a fixed number of plays, so-called the *budget*. In the first category, though we can partly control the number of rounds needed by fixing the required confidence, due to the unknown gap of the arms, the number of rounds is still unpredictable. Therefore, we consider the algorithm for the first in our general solution to be a fixed-budget setting as we can explicitly control the length of the standard MAB stage.

Many algorithms have been introduced to solve the Best Arm Identification problem in MAB in the fixed-budget setting. For example, Karnin [8] introduced the *Sequential-Halving* algorithm, where the budget is split into different phases and at each phase, the remaining arms are pulled in a uniform manner and the worst half of them will be ruled out. With a different approach, Audibert [2] proposed two algorithms. The first algorithm called *UCB-E*, is a highly exploring policy based on upper confidence bounds that explore arms for the whole budget and return the best empirical arm as the final output. Audibert also proposed a *sequential elimination* algorithm, namely *Successive-Reject* (SR), where the budget is also divided into phases as Karnin's method but only one arm is removed in each phase. The experimental result in [8] shows that SR has the lowest best-arm detection error of all methods.

Now we will introduce and analyse our algorithm for finding the m best arms in MAB in the fixed-budget setting, based on the idea of Successive-Reject, with an adaptation that instead of returning a single best arm, our algorithm's output is a set of m best arms. The algorithm is called *SR Best Arm Identification* and formally described in Algorithm 5.1.

Algorithm 5.1: SR Best Arms Identification

```
Input parameters: Number of arms in the output $m < K, m \in \mathbb{Z}_+$; time
  budget $n > K, n \in \mathbb{Z}_+$
Initialisation: $\mathcal{A}_1 = \{1, 2, \dots, K\}$; $S(m) = \frac{m}{m+1} + \sum_{i=m+1}^{K} \frac{1}{i}$
  $n_p = \lceil \frac{1}{S(m)} \frac{m+n-K}{K+1-p} \rceil$; $n_0 = 0$; $\lceil . \rceil$ is the ceiling function.
for phase $p = \{1, 2, \dots, K-m\}$ do
    (1) For each $i \in \mathcal{A}_p$, play arm $i$ for $n_p - n_{p-1}$ rounds.
    (2) Set $\mathcal{A}_{p+1} = \mathcal{A}_p \setminus \arg\min_{i \in \mathcal{A}_p} \hat{X}_{i,n_p}$ where $\hat{X}_{i,n_p}$ is the mean reward
      of arm $i$ after (1).
end for
Output: set of arm indices in $\mathcal{A}_{K-m}$ as the determined $m$ best arms.
```

Informally, the algorithm divides the budget of n rounds into $K - m$ phases. At the end of each phase, the arm with the lowest empirical mean is dismissed. During the next phase, each surviving arm is selected for the same number of rounds. The recommended final set is the arms that have not been dismissed after the last phase.

Lemma 5.1. *The successive-Reject Best Arms Identification algorithm's total number of rounds does not exceed the given budget of n pulls.*

Proof. At the end of every phase, we would have one arm being totally played for n_p rounds and removed. More precisely, one arm, which is removed after phase 1, is pulled for $n_1 = \lceil \frac{1}{S(m)} \frac{m+n-K}{K} \rceil$ times, one arm, which is removed after phase 2, is pulled for $n_2 = \lceil \frac{1}{S(m)} \frac{m+n-K}{K-1} \rceil$ times, and so on until the last phase, where there are $m+1$ arms, each of which is pulled for $n_{K-m} = \lceil \frac{1}{S(m)} \frac{m+n-K}{m+1} \rceil$ rounds. Therefore, the total number of rounds used for the algorithm is calculated as:

$$
\begin{aligned}
n_{total} &= n_1 + n_2 + \cdots + (m+1)\cdot n_{K-m} \\
&\leq (K-m) + \frac{m+n-K}{S(m)}\left(\frac{1}{K} + \frac{1}{K-1} + \cdots + \frac{1}{m+1} + \frac{m}{m+1}\right)\frac{m+n-K}{S(m)} \\
&= (K-m) + \frac{m+n-K}{S(m)} \cdot \left(\frac{m}{m+1} + \sum_{i=m+1}^{K} \frac{1}{i}\right) \\
&= (K-m) + (m+n-K) \\
&= n
\end{aligned}
$$

The inequality is derived from the inequality of the ceiling function: $\lceil x \rceil \leq x + 1$. The second equality is from the definition of $S(m) = \left(\frac{m}{m+1} + \sum_{i=m+1}^{K} \frac{1}{i}\right)$.

□

Despite containing 'Best Arms Identification' in the name, our algorithm's purpose does not exclusively focus on finding the optimal arms. On the other hand, here we aim at eliminating the bad arms out of consideration in order to reduce the number of available arms, which is to solve the discussed issues of MAB with a large number of arms.

5.3 Third research question

After presenting the problem of MAB with a large number of arms and proposing our solution, including the SR Best Arms Identification algorithm, we will now move on to our third research question, which will be explored in the following chapter.

RQ3) How does the arm reduction method improve the performance of the standard MAB algorithms in the extreme case of numerous arms?

The answer to this question will tell the effect of our arm reduction solution on the ultimate goal of maximising the cumulative reward in MAB.

Chapter 6

Arm Reduction Evaluation

This chapter presents the outcomes of our experiments, with the objective of addressing the research question RQ3: ”How does the arm reduction method improve the performance of the standard MAB algorithms in the extreme case of numerous arms?”. We will start by describing our experiment settings, including two investigated cases of the MAB problem with a large number of arms and the parameters for our arm reduction phase. Finally, we will present the obtained results and our findings.

6.1 Experiment settings

In our experiments with the proposed arm reduction method, we consider two extreme cases of the MAB problems as follows.

- **40-arm bandit**: We revisit the MAB problem that we introduced in Chapter 4, where we have 40 arms, in which there are 5 better arms with close mean rewards, including the actual optimal arm.
- **100-arm bandit** We raise the number of arms to 100 with 10 competitive arms.

Different from the previous experiments, now we set the time horizon T to be 3000 rounds. To reduce the number of arms in the 40-arm bandit and 100-arm bandit problems, we allocate respectively 400 rounds and 600 rounds for the first phase, which aims to determine the 10 best arms in the 40-arm bandit problem and 25 best arms in the 100-arm bandit problem.

In these experiments, we only consider two standard MAB strategies for the rest of the time horizon, Thompson Sampling and ε-greedy with $\varepsilon = 0.1$. Moreover, the history of pulling from the first phase will also be transferred to the next phase to be continued by the standard algorithms. As the previous experiments presented in Chapter 4, each game is repeated for 1000 times to see the variability of the method's performance. In the next section, we will provide the outcomes from the experiments, followed by a discussion of our findings regarding the results.

6.2 Results

6.2.1 40-arm bandit

In Figure 6.1, we present the distribution of the cumulative rewards of Thompson Sampling and ε-greedy with and without incorporating arm reduction solution.

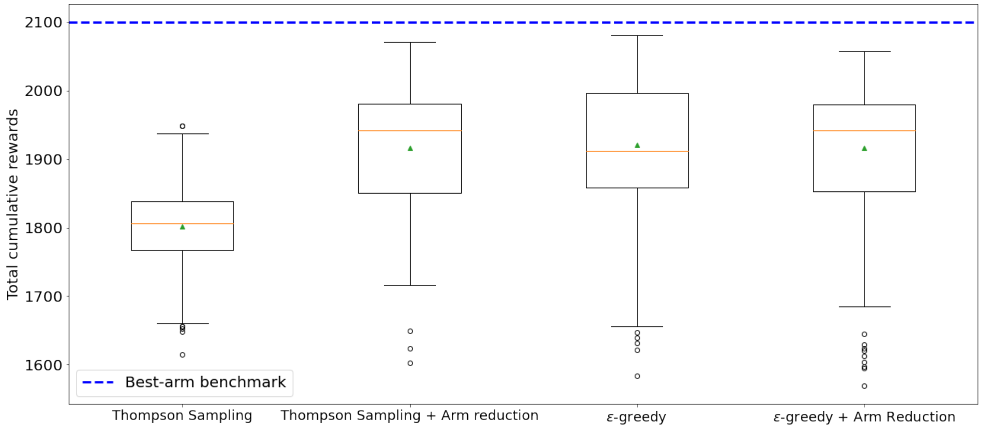


Figure 6.1: The cumulative rewards after 3000 rounds with 1000 repetitions

We can see that arm reduction barely makes an improvement to ε-greedy, however, significantly enhances the performance of Thompson Sampling.

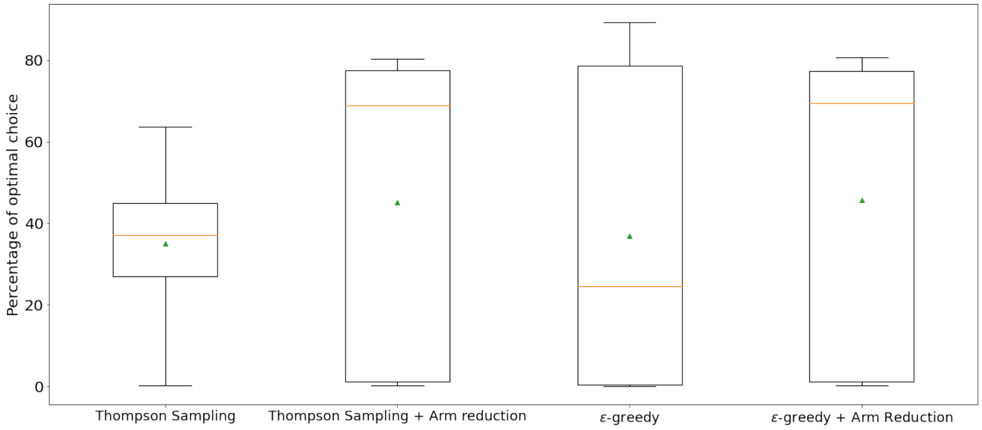


Figure 6.2: The percentage of optimal choice after 3000 rounds with 1000 repetitions

In terms of optimal choice percentage, which is presented in Figure 6.2, averagely, incorporating arm reduction clearly increases the chance of the actual optimal arm selected. On the other hand, the value is much more inconsistent for Thompson Sampling.

6.2.2 100-arm bandit

Figure 6.3 and 6.4 demonstrate the cumulative rewards and optimal choice percentage of the approaches after 3000 rounds.

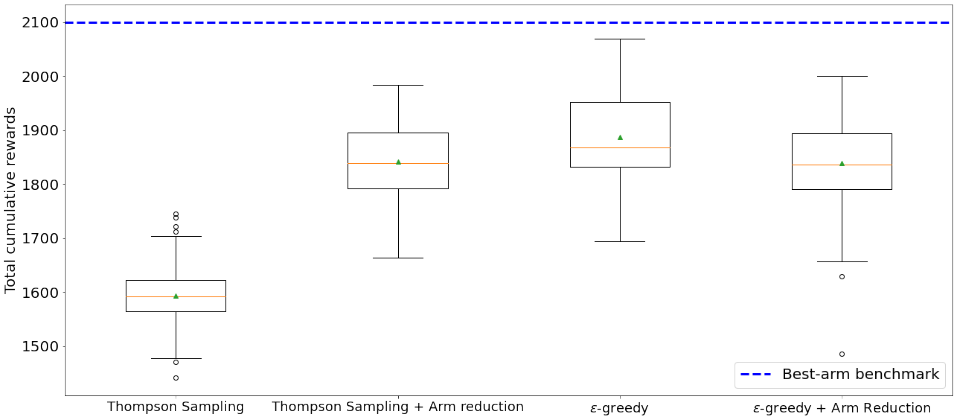


Figure 6.3: The cumulative rewards after 3000 rounds with 1000 repetitions

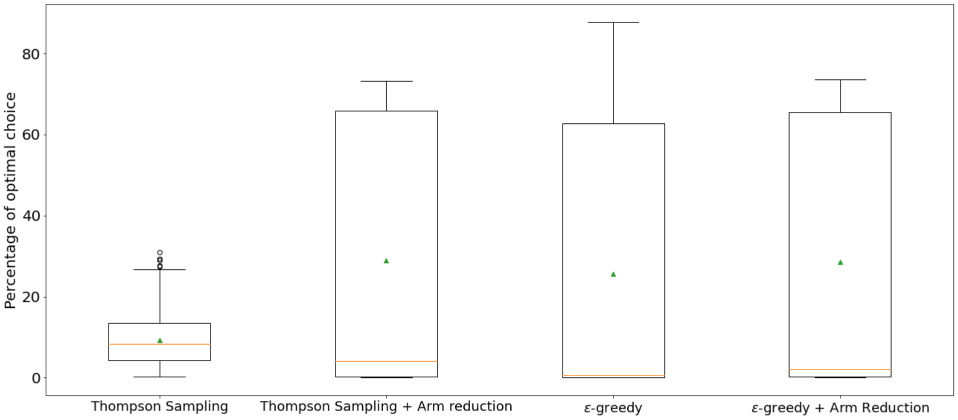


Figure 6.4: The percentage of optimal choice after 3000 rounds with 1000 repetitions

Different from the problem of 40 arms, doing arm reduction worsens the cumulative reward of ε-greedy though increases the optimal arm pulling rate. On the other hand, the method still boosts the overall performance when combined with Thompson Sampling.

6.3 Discussion

Firstly, looking at the experimental results of Thompson Sampling in the two scenarios, we can see that its performance was significantly improved by incorporating the arm reduction solution, in terms of both the cumulative reward and the optimal choice percentage. However, the collected results of these criteria were varied and less stable, especially with the optimal choice percentage. This can be explained by the added randomness from the Successive-Reject algorithm, where the output set of arms is likely to be different after each run, and the error of wrong elimination of the actual optimal arm can happen occasionally, which makes the optimal choice percentage to be very low at several repetitions of the experiments. However, even in that case, the arm reduction algorithm has successfully dismissed a large number of bad arms and helped increase the overall cumulative reward.

On the other hand, though increasing the percentage of the optimal arm selected, our arm reduction solution only made an inconsiderable improvement to ε-greedy's cumulative reward in the 40-arm bandit setting while it worsened the performance in the 100-armed bandit case. The observed results in previous sections might indicate that our arm reduction method is not effective enough to improve the algorithm. However, the outcomes from the experiments in Chapter 4 reveal that the performance of ε-greedy did not decrease as much as Thompson Sampling when the number of arms increased, hence, the room for improvement for ε-greedy is relatively smaller. Therefore, it requires a better choice of parameters, i.e. the budget for arm reduction algorithm and the number of output arms, which are not investigated in this project due to time limitations. We believe that if these parameters are optimally chosen, we might observe clear improvements for the ε-greedy algorithm.

Chapter 7

Conclusions and Future Work

In this chapter, we conclude the content of our report, including the key findings of the investigations into the two research questions presented in Chapter 3 and 5, discuss the limitation of our project and pose potential future research directions.

7.1 Conclusions

The report explored various strategies for balancing exploration and exploitation in multi-armed bandit (MAB) problems. In the initial chapter, we provided a background review of RL and MAB, including their key concepts such as rewards, exploration-exploitation trade-off, action-value functions, and regret. Subsequently, we presented the stochastic MAB problem featuring Bernoulli reward as the model for our study. Within the same chapter, we gave the answer to the first research question, which is:

RQ1) What are the potential solution candidates to handle the exploration-exploitation trade-off in MAB problem with Bernoulli reward signal?

We examined two well-established algorithms, namely ε-greedy and Thompson Sampling and proposed a novel method, the *Surprise-based Exploration policy*, that incorporates a surprise-based intrinsic reward with the external reward from the environment to promote exploration in MAB. After that, we posed our second research question:

RQ2) What is the empirical performance of the solution candidates in different scenarios of the MAB environment?

This question concerns the empirical performance of the proposed MAB algorithms, especially the effect of surprise-based exploration policy on the ε-greedy algorithm. To answer this question, we deployed experiments with three MAB instances of 4 arms, 10 arms and 40 arms. According to the results, when the number of arms is relatively low, Thompson Sampling demonstrated superior performance compared to other approaches and our proposed exploration policy yielded some improvement in the performance of ε-greedy. However, when we increase the number of arms, the cumulative rewards obtained from Thompson Sampling plummeted largely and the improvement from surprise-based exploration policy became smaller and to a point caused a negative effect on ε-greedy.

Upon reviewing the results of the 40-arm experiment, we recognized an issue that arises in MAB with a large number of arms. We delved deeper into the problem by first giving a brief overview of the problem with potential causes. Subsequently, we suggested a solution to solve the problem by reserving a number of time steps in the early stage of the time horizon to prune the worst arms, using the proposed algorithm *SR Best Arms Identification*, before feeding the remaining arms into a standard MAB strategy. Ultimately, we formulated our third research question:

RQ3) How does the arm reduction method improve the performance of the standard MAB algorithms in the extreme case of numerous arms?

In order to address this question, we perform our proposed solution in two scenarios of MAB with 40 arms and 100 arms and compare the results to the original method, which is performing the standard MAB algorithms for the whole time horizon. Observing the outcomes of the experiments, we found that Thompson Sampling largely benefited from the arm reduction method. On the other hand, the method is not shown to be effective when incorporating with ε-greedy.

7.2 Future Work

Due to constraints on time, there were various factors that we were unable to explore but could be considered in future research. Firstly, for the first part of our investigation on the MAB approaches, the value of parameter η as the exploration urge in our proposed Surprise-based Exploration policy is not well investigated and could affect the algorithm's performance. In future work, this could be solved either by developing a theoretical regret bound for the method and determining the optimal η through optimisation or by conducting experiments to assess the impact of various η values and identifying how to select the parameter for different scenarios.

Secondly, regarding our solution to MAB with numerous arms, we gave a discussion about the challenge of time step distribution of the two phases and the selection of m as the number of arms in the output of the first phase and how they could affect the performance of the approach. However, we did not make an investigation into this aspect, which can be a direction of future work. We also acknowledge that explicitly dividing the time horizon and choosing the number of arms to dismiss is a difficult problem since it depends on many variables, such as the length of the time horizon, the number of arms and the gaps of the arms, which is unknown to the agent. Therefore, in future work, we can consider an adaptive policy where the exploration is highly prioritised in the beginning but decreased over time and suboptimal arms are gradually removed at some confidence level.

Finally, our MAB problem formulation with Bernoulli reward signal is not applicable to real-world problems, where the rewards can not be simply modelled as 1 and 0. Hence, an investigation into other reward distributions or reward ranges of values is a good extension for the future project.

Bibliography

[1] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 39.1–39.26, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

[2] Jean-Yves Audibert and Sébastien Bubeck. Best Arm Identification in Multi-Armed Bandits. In *COLT - 23th Conference on Learning Theory - 2010*, page 13 p., Haifa, Israel, June 2010.

[3] Nicolò Cesa-Bianchi and Paul Fischer. Finite-time regret bounds for the multiarmed bandit problem. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 100–108, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[4] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[5] Thore Graepel, Joaquin Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. pages 13–20, 06 2010.

[6] Ole-Christoffer Granmo. Solving two-armed bernoulli bandit problems using a bayesian learning automaton. *International Journal of Intelligent Computing and Cybernetics*, 3:207 – 234, 08 2010.

[7] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration, 2017.

[8] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1238–1246, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[9] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson Sampling: An Asymptotically Optimal Finite Time Analysis. In *ALT 2012 - International Conference on Algorithmic Learning Theory*, volume 7568 of *ALT 2012: Algorithmic Learning Theory*, pages 199–213, October 2012.

[10] Duc-Huy Le, Hai-Anh Tran, and Sami Souihi. A reinforcement learning-based solution for intra-domain egress selection. In *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*, pages 1–6, 2021.

[11] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 2010.

[12] Benedict C. May. Simulation studies in optimistic bayesian sampling in contextual-bandit problems. 2011.

[13] Benedict C. May, Nathan Korda, Anthony Lee, and David S. Leslie. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(67):2069–2106, 2012.

[14] Herbert Robbins. Some aspects of the sequential design of experiments. 1952.

[15] Steven L. Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[16] Satinder Singh, Richard L. Lewis, Andrew G. Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

[17] Aleksandrs Slivkins. Introduction to multi-armed bandits. *CoRR*, abs/1904.07272, 2019.

[18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* A Bradford Book, Cambridge, MA, USA, 2018.

[19] Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, page 1587–1594, New York, NY, USA, 2013. Association for Computing Machinery.

[20] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[21] Joannès Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In João Gama, Rui Camacho, Pavel B. Brazdil, Alípio Mário Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005*, pages 437–448, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[22] Sofía S. Villar, Jack Bowden, and James Wason. Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges. *Statistical Science*, 30(2):199 – 215, 2015.

[23] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, UK, May 1989.

Appendix

This appendix provides the theorems that we used for our theoretical analysis of the Multi-armed Bandit algorithm.

A.1 Corollary of Hoeffding inequality

Let $X_1, \dots, X_n \sim F$ be $\mathbb{R}$-valued random variables such that $\mathbb{P}(X_i \in [a,b]) = 1$, then for any $\varepsilon > 0$, we get for $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$,

$$\mathbb{P}(\bar{X}_n - \mathbb{E}[\bar{X}_n] \geq \varepsilon) \leq \exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right)$$

$$\mathbb{P}(|\bar{X}_n - \mathbb{E}[\bar{X}_n]| \geq \varepsilon) \leq 2\exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right)$$

A.2 Union bound

In probability theory, **union bound**, also known as **Boole's inequality**, states that for a finite or countable set of events, the probability of at least one of them occurring is not higher than the sum of the individual probabilities of each event.

Formally, for a countable set of events $A_1, A_2, A_3, \dots$ we have

$$\mathbb{P}\left(\bigcup_i A_i\right) \leq \sum_i \mathbb{P}(A_i)$$

A.3 Proof of Lemma 3.1

Recall from Remark 3.2 that Δ_a is the gap of arm a. Let a sequence $Y_1, Y_2, \dots, Y_n$ with $Y_i = X_{a,i} - X_{a^*,i}$ where $X_{a,i}$ is the reward of arm a at its i-th selected round. Therefore Y_i is a random variable with support $[-1,1]$ and $\mathbb{E}[Y_i] = -\Delta_a$. Let $\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$

Applying a corollary of Hoeffding inequality (see Appendix A.1) , we get

$$\mathbb{P}(\hat{\mu_k} - \hat{\mu_*}) = \mathbb{P}(\overline{Y} - \mathbb{E}[Y] \geq \Delta_k) \leq 2\exp\left(-\frac{n\Delta_k^2}{2}\right)$$

Using union bound (see Appendix A.2), we have

$$\mathbb{P}(\exists k\, \hat{\mu_k} - \hat{\mu_*} \geq 0) \leq \sum_k \mathbb{P}(\hat{\mu_k} - \hat{\mu_*} \geq 0) \leq \min\left(1, \sum_k \exp\left(-\frac{n\Delta_k^2}{2}\right)\right)$$

The second inequality above contains a min function because a probability could not exceed 1 by definition.

A.4 Proof of Theorem 3.2

At round t, the expected number of exploration rounds (with exploration probability ε) is εt hence the expected number of exploration rounds $\mathbb{E}[N_{a,t}] = \frac{\varepsilon t}{K}$ for all arms. Let $\hat{\mu_a}$ be the empirical mean reward of arm a up to round t. Using a corollary of Hoeffding inequality (see Appendix A.1) for a sequence of Bernoulli random variables, we have for every arm a

$$\mathbb{P}(|\hat{\mu_a} - \mu_a| \geq r) \leq 2\exp\left(\frac{-2\varepsilon tr}{K}\right)$$

Choosing $r = \sqrt{\frac{2K\log t}{\varepsilon t}}$, we have

$$\mathbb{P}(|\hat{\mu_a} - \mu_a| \geq r) \leq \frac{2}{t^4} \tag{A.1}$$

Using the union bound (see Appendix A.2), we have

$$\mathbb{P}\left(\bigcup_a |\hat{\mu_a} - \mu_a| \geq r\right) \leq \frac{2K}{t^4}$$

Hence

$$\mathbb{P}\left(\forall_a\, |\hat{\mu_a} - \mu_a| \leq r\right) \geq 1 - \frac{2K}{t^4} \tag{A.2}$$

In words, this implies the empirical mean for every arm is bounded in range $[\mu_a - r, \mu_a + r]$ with probability at least $1 - \frac{2K}{t^4}$, which is very large even with a small t. Therefore, we can ignore the case where there is at least one of the empirical means falling out of that range.

Remark A.1. In fact, it is likely that an arm is expected to be pulled more than $\frac{\varepsilon t}{K}$ times since we also have $(1 - \varepsilon)t$ exploitation rounds excluded in this analysis. However, the probability bound in (A.1) is larger if the exploitation rounds are considered, therefore (A.2) still holds.

Now we divide the situation into two cases.

Case 1. If this round is an exploration round, with probability of ε. In this case, as the arms are chosen randomly with equal probability, the expected regret for this case is $\frac{1}{K}\sum_{a\in\mathcal{A}} \Delta_a$

Case 2. If this round is an exploitation round, with probability $1 - \varepsilon$, and the actual optimal arm is chosen, i.e. $a_t = a^*$, the expected regret is 0 as the best arm is played. On the other hand, if a suboptimal arm a is played instead of a^*, i.e. $\hat{\mu_a} > \hat{\mu^*}$ the following holds

$$\mu^* + r \geq \hat{\mu_a} > \hat{\mu^*} \geq \mu^* - r \Leftrightarrow \Delta_a \leq 2r$$

Using Lemma 3.1 with $n = \frac{\varepsilon t}{K}$, we have the probability of one of the suboptimal arms being chosen as

$$\mathbb{P}(\exists_k \, \hat{\mu_k} \geq \hat{\mu_*}) \leq \min\left(1, \sum_k \exp\left(-\frac{\varepsilon t \Delta_k^2}{2K}\right)\right)$$

Hence the expected regret, in this case, is bounded as

$$\mathbb{P}(\exists_k \, \hat{\mu_k} \geq \hat{\mu_*}) \cdot \Delta_a \leq \min\left(1, \sum_{a\in\mathcal{A}} \exp\left(-\frac{\varepsilon t \Delta_a^2}{2K}\right)\right) \cdot 2r$$

Combine two cases, we have the upper bound for the expected regret at round t

$$\begin{aligned}\mathbb{E}(R(t)) &= \mathbb{P}(Exploration) \cdot \text{Exploration-regret} + \mathbb{P}(Exploitation) \cdot \text{Exploitation-regret} \\ &\leq \frac{\varepsilon}{K} \sum_{a\in\mathcal{A}} \Delta_a + (1-\varepsilon) \min\left(1, \sum_{a\in\mathcal{A}} \exp\left(-\frac{\varepsilon t \Delta_a^2}{2K}\right)\right) \cdot 2r\end{aligned}$$